This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

Network Coding Optimization Based on Chemical Reaction Optimization

Bo Pan*, Albert Y.S. Lam†, *Member, IEEE*, and Victor O.K. Li*, *Fellow, IEEE*
*Department of Electrical and Electronics Engineering, The University of Hong Kong
†Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
Email: panbo@hku.hk, albertlam@ieee.org, vli@eee.hku.hk

***Abstract*—Network coding may improve network efficiency. However, it is not necessary to code every link to meet a given transmission rate. In this paper, we consider the NP-hard problem of minimizing the number of coding links of a network for a given target transmission rate. Chemical Reaction Optimization (CRO) is a general purpose metaheuristic, which have been demonstrated to be effective in many optimization problems. We adopt the CRO framework to develop an algorithm to solve this NP-hard problem. Simulation results show that CRO outperforms existing algorithms with two sets of test network topologies.**

***Index Terms*—network coding, optimization, evolutionary algorithm, chemical reaction optimization.**

I. Introduction

In traditional communication networks, the sources send data packets to the destinations via certain intermediate nodes (i.e. routers), which simply receive and forward data packets [1]. With this traditional routing, to increase the overall network throughput, we need to add additional links and nodes to the network, but this is not always feasible. It has been shown that by combining received information, network coding can effectively increase the throughput and the multicast rate without any topological change to the network [2][3][4].

Typical network coding techniques assume that network coding is performed at all the links of a network, but coding at most of the links is unnecessary in most situations. The excessive coding links only increases the computational cost without contributing to the network efficiency. For example, when network coding takes place at a link, we need to transform the optical signals into the corresponding electrical ones [5]. Therefore, this gives rise to the network coding optimization problem (NCOP): What is the minimal number of links for a certain network to achieve a target transmission rate? We are also interested in the results of the network protocols which determine how many links and which links need to be coded in the networks.

Fragouli *et al.* [6] derived an information flow decomposition algorithm, which can calculate the improvement of the overall throughput for a network when network coding takes place. Lanberg *et al.* [7] determined the maximum number of coding nodes needed to achieve the transmission rate in both acyclic and cyclic networks. They also demonstrated that NCOP is NP-hard. In [5], Bhattad *et al.* developed a new information flow model to derive algorithms for optimization problems involving network coding. Kim *et al.* is probably the first to use evolutionary algorithms to solve NCOP [1], where Genetic Algorithm (GA)-based method under an algebraic framework [8] was proposed. The chromosome encoding scheme is applied to the labeled line graph. They also improved their algorithm by changing the chromosome encoding strategy [9][10]. With the price of sacrificing some of the intermediate coding information, the search space is effectively reduced, and thus the efficiency greatly increased. Hu *et al.* proposed a GA-based algorithm with a new encoding scheme [11][12]. This method simply uses the current network topology to complete the chromosome encoding process.

Chemical Reaction Optimization (CRO) is a newly developed evolutionary algorithm to search for global optima for general optimization problems [13]. CRO is inspired by the phenomenon of chemical reactions, where molecules constantly collide with each other and the wall of the container, and they have the tendency to produce products with the lowest and most stable energy state. CRO models feasible solutions as molecules, and searches through the solution space for other possible better solutions with four pre-defined elementary reactions. Since the molecules in a chemical reaction tend to converge to the lowest energy state on the potential energy surface (PES), we will be able to obtain the global minimum solution by this analogy. CRO has been used to solve many optimization problems with desirable performance [14][15][16]. Inspired by the application of GA to NCOP in the previous work, we try to study if CRO can tackle the problem effectively.

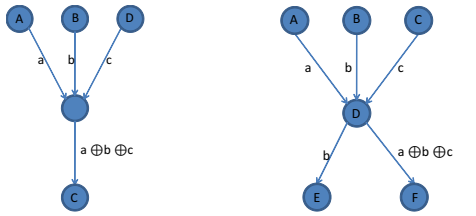
In this paper, we propose a CRO-based evolutionary algorithm to solve NCOP. We provide a new way to encode the network protocols, and design effective operators and constraint functions to search for the global optimum effectively. The simulation results show that CRO can outperform existing algorithms to this problem.

The rest of this paper is organized as follows. Section II gives the problem formulation. We explain our algorithm in details in Section III. In Section IV, we present our simulation results and compare CRO with existing algorithms. We conclude the paper in Section V.

II. Problem Formulation

In this paper, we consider a directed acyclic multicast network modeled as $G = (V, E)$, where V and E represent

978-1-4244-9268-8/11/$26.00 ©2011 IEEE



(a) Linear coding is performed at a coding node

(b) Linear coding is performed at Link DF

Fig. 1. Examples of linear coding

nodes and links, respectively. We assume each link transmits data at unit capacity. Multiple links are allowed between two different nodes, and thus, an arbitrary data rate between two nodes can be achieved by providing sufficient number of links in between. It is also possible that some of the links do not transmit any data based on the network coding scheme. For networks with multiple source nodes, we can add an imaginary source node to multicast data to the current multiple source nodes. Thus we always assume our network has one single source node S. The network transmission rate is denoted as R, which is the data rate that the source S transmits. R is said to be achievable if every sink D of the network can receive the data from S with rate R.

We call an intermediate node with multiple incoming links a merging node. Network coding happens at the merging nodes, which can combine the packets from its multiple incoming links and transmit the resultant packet to its outgoing links. Here we only consider the basic linear coding, which simply combines the data with the Exclusive-OR operation (See Fig. 1). This linear coding is enough for multicasting [4]. The maximum multicast rate R_{max} of a given network can be determined if all the merging nodes are coded [2]. Target rate must be $R \leq R_{max}$. Instead of the coding nodes, we are more interested in the coding links. The reason is that we can determine exactly how each link transmits data in the network by locating the coded links. Thus we need a network protocol to specify and indicate whether links should carry coded data from its incoming node.

The constraint is to achieve the target transmission rate under a certain network protocol which is a candidate solution to the optimization problem. To do this, we check at every sink D in the network if it can receive a whole set of information from the source S at rate R. Unlike [1], our implementation enables us to compare the data at each sink directly, without calculating the system matrix at each sink.

For a network with N possible coding links and the i^{th} possible coding node has n_i incoming links. If none of them is chosen, there will be $C_0^{n_i}$ possible combinations. If one is chosen to transmit, it is $C_1^{n_i}$. If two of them are coded, it is $C_2^{n_i}$, etc. Thus the total number of possible network protocols is

$$\prod_{i=1}^{N}(C_0^{n_i} + C_1^{n_i} + C_2^{n_i} + ... + C_{n_i}^{n_i}). \quad (1)$$

Langberg in [7] proved the problem of minimizing the number of coding nodes with any multiplicative factor is NP-hard, and in [1], NCOP for coding links is also shown to be NP-hard.

III. Algorithm Design

A. Proposed Approach

CRO is implemented in three stages: the initialization, the iterations, and the termination. In the initialization, we initialize two sets of data. The first set is for the network topology specification, including the number of links ($link_{No}$), the starting and ending nodes of each link i ($link_{Start}(i)$ and $link_{End}(i)$). We use $link_{Next}$ to indicate Link i is the $link_{Next}th$ incoming link of its ending node. $Link_{Max}(i)$ represents the number of incoming links of the starting node of Link i. We also need to specify the source and the sinks for the network. The other set of data is the algorithmic parameters of CRO. We apply the standard CRO framework [13] to develop the algorithm. Interested readers may refer to [13] for the details of CRO.

B. The Solution Structure and Encoding Scheme

A solution (i.e. molecular structure of a molecule) of the problem indicates which links of the network need to be encoded. Different solution representations will greatly affect the algorithm efficiency. A molecule in CRO is a vector containing $|E|$ integers, where $|E|$ is the number of links of the network. Each integer specifies how a certain link receives or routes data from previous incoming links. Our encoding scheme is similar to that in [11] with one major difference. [11] assumes that a coding node only combines two of its incoming signals[1], but in our algorithm, if the number of incoming links of a link is $link_{Max}$, we will use $link_{Max} + 1$ to indicate that the link is coded, and it combines signals from all its incoming links. It has been shown in [9] that as long as a link is coded, no matter how many links it combines, it can still satisfy the target transmission rate of the network. For example, Fig. 1(b) shows a network, within which Link AD, BD and CD transmit data a, b and c, respectively, with known $link_{Next}(AD)$=1, $link_{Next}(BD)$=2 and $link_{Next}(CD)$=3. If Link DE is encoded as 2 and Link DF is encoded as 4, DE transmits b and DF transmits the coded signal. With this encoding scheme and the network topology data set, we can deal with arbitrary network topologies and determine the information transmitted on each link. This will help us to conduct the feasibility check.

C. Operators and Constraint Function

1) Molecule generator: This operator is designed to generate new feasible molecules based on our solution structure. We randomly assign every element e of a molecule an integer r in the range of [1, $link_{Max}(e) + 1$] and then perform the feasibility check to see if the molecule is feasible. If it passes the feasibility check, the molecule will be used for the reactions afterwards. Otherwise, we will discard the

[1]Each node may have more than two incoming links, but it can only combine the data from two of the incoming links.

molecule and generate a new one. The pseudo code is shown in Algorithm 1.

Algorithm 1 MOLGEN(m)

```
m is feasible=False
while m is feasible=False do
    for all Elements e in m do
        randomly generate a real number r
        r ≤ (link_Max(e) + 1)
        e = r
    end for
    conduct feasibility check for m
end while
```

2) On-wall Ineffective Collision: This operator is used to generate a solution in the neighborhood of a given molecule. To do this, we randomly choose a coding Link p with $link_{Max}(p) > 1$ and randomly assign another possible value to it. For example, for a network of 5 links whose 2nd and 3rd links are possible coding links, with $link_{Max}(2) = 2$ and $link_{Max}(3) = 3$. Here we randomly choose to modify the 3rd link, which originally carries data from its 2nd precedent link, to transmit coded data, as follows:

$$(1, 2, \underline{2}, 1, 1) \rightarrow (1, 2, \underline{4}, 1, 1) \tag{2}$$

Then we check the feasibility of the resultant solution. The pseudo code is shown in Algorithm 2.

Algorithm 2 ON WALL(m)

```
randomly choose a possible coding link p
new molecule m' is feasible=False
while new molecule m' is feasible=False do
    randomly generate a real number r
    r ≤ (link_Max(p) + 1)
    p = r
    conduct feasibility check for m'
end while
```

3) Decomposition: This operator is used to produce two new molecules (m_1' and m_2') far away from a given one. m_1' is generated by assigning a new solution (by the molecule generator). Based on this molecule, we obtain m_2' by assigning every element $e_{m2'}$ of m_2' with $link_{Max}(e_{m2'}) + 2 - e_{m1'}$ to it. With the example used above, we modify both coding links, namely, the 2nd and the 3rd link, which originally carry data from their 2nd precedent link, respectively. The 2nd link of m_1' is modified to carry coded data, and the 2nd link of m_2' is changed to carry coded data from its 1st precedent link. The 3rd link of m_1' remains unchanged, and the 3rd link of m_2' is modified to carry data from its 3rd precedent link, as follows:

$$(1, \underline{2}, \mathbf{2}, 1, 1) \longrightarrow (1, \underline{4}, \mathbf{2}, 1, 1) + (1, \underline{1}, \mathbf{3}, 1, 1) \tag{3}$$

Then we check the feasibility of the resultant solutions. The pseudo code is shown in Algorithm 3.

4) Inter-molecular Ineffective Collision: This operator takes two molecules (m_1 and m_2) to produce two new molecules (m_1' and m_2'). This operator changes a possible coding link of the two molecules. We randomly pick a possible

Algorithm 3 DECOMPOSITION (m)

```
copy m to m_1' and m_2'
m_1', m_2' are feasible=False
while m_1' is feasible=False do Do
    for all Elements e_{m1'} in m_1' do
        generate a real possible value r
        r ≤ (link_Max(e_{m1'}) + 1)
        e_{m1'}=r
    end for
    conduct feasibility check for m_1'
end while
while m_2' is feasible=False do
    for all Elements e_{m2'} in m_2' do
        e_{m2'} = link_Max(e_{m2'}) + 2 − e_{m1'}
    end for
    conduct feasibility check for m_2'
end while
```

coding link p. For m_1, we do the same change as Algorithm 2 to obtain m_1', and we change the corresponding element of m_2 (p_{m2}) into $link_{Max}(p_{m2'}) + 2 - p_{m1'}$. We still use the same network in Algorithm 2 as an example. The 3rd link is randomly chosen to be modified. The 3rd link of m_1 is changed to transmit coded data which originally transmits data from the 2nd incoming link. The same link of m_2 is assigned with $link_{Max}(3) + 2 - p_{m1'}$, which means it now transmits data from the 1st precedent link, shown as follows:

$$(1, 2, \underline{2}, 1, 1) + (1, 2, \mathbf{2}, 1, 1) \longrightarrow (1, 2, \underline{4}, 1, 1) + (1, 2, \mathbf{1}, 1, 1) \tag{4}$$

Then we check the feasibility of the resultant solutions. The pseudo code is shown in Algorithm 4.

Algorithm 4 INTERMOL (m_1,m_2)

```
randomly choose a possible coding link p
m_1', m_2' are feasible=False
while m_1' is feasible=False do
    generate a real possible value r
    r ≤ (link_Max(p_{m1'}) + 1)
    p_{m1'}=r
    conduct feasibility check for m_1'
end while
while m_2' is feasible=False do
    p_{m2'} = link_Max(p_{m2'}) + 2 − p_{m1'}
    conduct feasibility check for m_2'
end while
```

5) Synthesis: This operator takes two molecules m_1 and m_2 to generate one new molecule m' by randomly taking the elements from m_1 and m_2 for every element in m'. For example,

$$(\underline{1}, 2, 2, \underline{1}, 1) + (1, \mathbf{3}, \mathbf{2}, 2, \mathbf{1}) \longrightarrow (\underline{1}, \mathbf{3}, \mathbf{2}, \underline{1}, \mathbf{1}) \tag{5}$$

We check the feasibility of the new molecule. The pseudo code is shown in Algorithm 5.

6) Feasibility Check: Feasibility check (constraint function) is very important to CRO to solve NCOP. For any newly generated molecule, we need to check whether each sink D can receive the whole set of data sent from the source. Our constraint function contains two parts. In the first part, we try to check if every sink can receive data. To do this, we

Algorithm 5 SYNTHESIS (m')

```
m' is feasible=False
while m' is feasible=False do
    for all Elements in e in m' do
        generate a real r between 0 and 1
        if r < 0.5 then
            e =counterpart in m_1
        else
            e =counterpart in m_2
        end if
    end for
end while
```

Algorithm 6 FEASIBILITY CHECK (m)

```
any link contains new data=True
while any link contains new data=True do
    for all Elements e in m do
        if e is coding link=True then
            e combines data from its previous links
        else
            e routes data from its previous links
        end if
    end for
    check if e contains new data
end while
for all Sinks d in m do
    if d receives whole set of information from source s=True then
    return True;
    else return False;
    end if
end for
```

transmit data from the source to the sinks through the network specified with the link coding situation corresponding to a molecule. The second part is to verify whether each sink D can decode the received information and recover the original whole set of data from the source S. The pseudo code is shown in Algorithm 6.

D. Stopping Criteria

A function evaluation (FE) refers to one call to the objective function. This algorithm will terminate when a certain number of FEs are reached.

IV. SIMULATION RESULTS

A. Experiment Setup

We prepare two sets of networks for evaluation. The first set is a regular network topology which is widely tested by other existing network coding optimization algorithms [1][9][11]. The details of the first test set can be found in [1] and [9].

TABLE III
PERFORMANCE OF THE ALGORITHMS FOR TEST SET 2.

Network	Randomized 1			Randomized 2		
	Best	Avg.	Std.	Best	Avg.	Std.
Minimal 1	0	1.35	-	0	1.85	-
Minimal 2	0	1.85	-	0	1.95	-
bit-link-state NCGA	0	1.20	-	0	1.05	-
CRO	0	1.00	0.32	0	0.50	0.59

There are four network topologies in this set, consisting of 3 copies, 7 copies, 15 copies, and 31 copies of the basic structure. The transmission rate is 2 which can be achieved with no coding on any links [11].

In order to test whether CRO can work well in other more practical scenarios, we produce the second set of networks for testing. It contains two uniformly randomly connected acyclic graphs, constructed according to [17]. Both networks have one single source. The first network has 20 nodes, 80 links, and 12 sinks with transmission rate equal to 4. The second network has 40 nodes, 120 links, and 12 sinks, with transmission rate equal to 3. The parameters for the test network sets are listed in Table I.

For the first set, we compare our results with two greedy algorithms, Fragouli's Minimal 1 [6] and Langberg's Minimal 2 [7], and other existing evolutionary algorithms including bit-link-state (BLS) NCGA [1], block-transmission-state (BTS) NCGA [9][10], GA [11]. For fair comparisons, the maximum allowed FEs are set to 15000 and 45000, respectively, according to [9] and [11]. The performance is evaluated with the average results of 30 and 20 trials for the sets, respectively. CRO's parameter settings are Initial Population Size = 10, Initial Energy Buffer = 0.0, Initial KE = 800.0, Molecular Collision Rate = 0.2, KE Loss Rate = 0.2, Decomposition Threshold = 300, and Synthesis Threshold = 10. There are no general rules to determine these parameters, and thus, we set the combination of parameter values by trial and error.

B. Experiment Results and Evaluation

Table II shows the results for the first test set. In terms of the best solution, CRO finds the global minimum for each case and performs as well as the other evolutionary algorithms. In terms of the average, CRO achieves similar results as BTS-NCGA and GA, and significantly outperforms the other algorithms. The standard deviation for GA is not listed because it is not provided in [11].

Table III gives the results for the second test set. CRO outperforms the two greedy algorithms and BLS NCGA. All the algorithms can achieve the same best results, while CRO gives the best average performance. GA is not shown here because there are no simulation results on randomized networks in [11], while BTS NCGA is also not given because it was tested on a different set of networks in [9].

From the above, we can see that CRO is very competitive when compared with other algorithms. In fact, CRO can achieve the similar simulation results with fewer FEs (See the convergence graph in Fig. 2). There are two reasons for the high convergence speed of CRO. The first is the ability for CRO to jump out of a local minimum and quickly search other possible better results. The second is due to the efficient encoding scheme, which greatly reduces the search space.

V. CONCLUSION

In this paper, we successfully apply an evolutionary algorithm CRO to solve NCOP. CRO has the following advantages in solving the problem: (1) CRO can show both the minimum

TABLE I
NETWORK DETAILS

Network	Test Set 1				Test set 2	
	3 copies	7 copies	15 copies	31 copies	randomized 1	randomized 2
Molecule length	30	70	150	310	80	120
Search space size (log_{10})	7.6339	19.0849	41.9876	87.7903	30.934	29.7031
Number of Nodes	19	43	91	187	20	40
Number of Links	30	70	150	310	80	120
Number of Sinks	4	8	16	32	12	12
Transmission Rate	2	2	2	2	4	3

TABLE II
PERFORMANCE OF THE ALGORITHMS FOR TEST SET 1.

Network	3 copies			7 copies			15 copies			31 copies		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
Minimal 3	3.00	0.00	7	7.00	0.00	15	15.00	0.00	31	31.00	0.00	
Minimal 2	0	2.15	0.86	2	4.70	1.25	7	11.60	1.65	28	52.80	2.66
BLS NCGA	0	0.93	0.69	0	2.20	1.27	3	5.57	1.55	12	12.43	2.37
BTS NCGA	0	0.00	0.00	0	0.00	0.00	0	0.17	0.38	0	1.03	0.81
GA	0	0.00	-	0	0.00	-	0	0.00	-	0	0.15	-
CRO	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.10	0.30

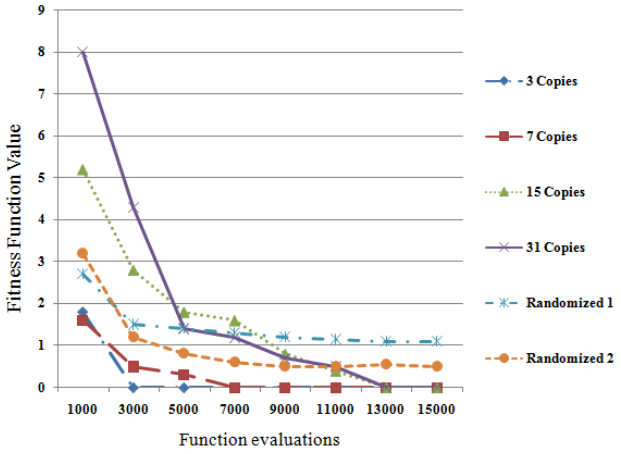


Fig. 2. The convergence graph of CRO for different test samples

number of coding links and the coding scheme of every link in the network; (2) with known network topologies, CRO simplifies the process of feasibility check and molecule encoding, without incurring the extra steps of calculating the system matrix at each sink or establishing a corresponding labeled line graph during encoding; (3) CRO can deal with pragmatic multicast scenarios. In the simulation, CRO outperforms the existing algorithms, in terms of generality and solution quality, and it has a high convergence rate.

REFERENCES

[1] M. Kim, C.W. Ahn, M. Medard and M. Effros, "On minimizing network coding resources: An evolutionary approach," *in Proc. NetCod. 2006*, Boston, 7 April 2006.

[2] R. Ahlswede, N. Cai, S.Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no.4, pp. 1204-1216, 2000.

[3] C. Fragouli, J.Y.L. Boudec, and J. Widmer, "Network Coding: An Instant Primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, No.1, Jan. 2006.

[4] S.-Y.R. Li, R.W. Yeung, and N. Cai. "Linear network coding," *IEEE Trans. on Inform. Theory*, vol. 49, pp. 371-381, 2003.

[5] K. Bhattad, N. Ratnakar, R. Koetter, and K. R. Narayanan, "Minimal network coding for multicast," *in Proc. IEEE ISIT '05*.

[6] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inform. Theory*, vol.52, no.3, pp. 829-848, 2006.

[7] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *Proc. IEEE ISIT '05*.

[8] R. Koeter and M. Medard, "An algebraic approach to network coding," in *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782-795, 2003.

[9] M. Kim, V. Aggarwal, U.M. O'Reilly, M. Medard and W. Kim, "Genetic representation for evolutionary minimization of network coding resources," *Applications of Evolutionary Computing, 2007*.

[10] M. Kim, M. Medard, V. Aggarwal, U.M. O'Reilly, W. Kim, C.W. Ahn and M. Effros, "Evolutionary approaches to minimizing network coding resources," *in Proc. the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007)*, Anchorage, AK, May 2007.

[11] Hu X.B., M.S. Leeson and E.L. Hines, "An effective Genetic Algorithm for the Network Coding Problem," *in the 2009 IEEE Congress on Evolutionary Computation (CEC2009)*, Norway, 18-21 May 2009.

[12] Hu X.B., M.S. Leeson and E.L. Hines, "Dynamic Network Coding Problem: An Evolutionary Approach," *WiCOM2009*, Beijing, China, 23-25, Sep, 2009.

[13] A.Y.S. Lam and V.O.K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Trans. Evol. Computation*, vol. 14, no. 3, pp. 381-399, Jun. 2010.

[14] A.Y.S. Lam, J. Xu, and V.O.K. Li, "Chemical reaction optimization for the population transition problem in peer-to-peer live streaming," *in Proc. Congr. Evol. Comput.*, Barcelona, Spain, pp. 1429-1436, July 2010.

[15] J. Xu, A.Y.S. Lam, and V.O.K. Li, "Chemical reaction optimization for task scheduling in grid computing," *IEEE Trans. Parallel Distrib. Systems*, accepted for publication.

[16] A.Y.S. Lam, V.O.K. Li, "Chemical Reaction Optimization for Cognitive Radio Spectrum Allocation," *in Proc. GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, 6-10 Dec. 2010.

[17] G. Melancon and F. Philippe, "Generating connected acyclic digraphs uniformly at random," *Inf. Process. Lett.*, vol. 90, no. 4, pp. 209-213, 2004.